Automatic Error Detection Method for Japanese Particles

Hiromi Oyama

Abstract:
In this article, I propose an approach for detecting appropriate usage models of case particles in the writings of Japanese Second Language learners (JSL) in order to create a Japanese automatic error detection system. As learner corpora are receiving special attention as an invaluable source for the educational feedback to improve teaching material and methodology, automatic methods of error analysis have become necessary to facilitate the development of learner corpora. Particle errors account for a substantial proportion of all grammatical errors by JSL learners and discourage the readers from understanding the meaning of a sentence. To address this issue, I trained Support Vector Machines (SVMs) to learn correct patterns of case particle usages from a Japanese newspaper text corpus. The result differs according to the kind of the particle. The object marker "wo (を)" has the best score of 81.4%. Applying the "wo (を)" model to detect wrong use of the particle, the result shows 92.6% for precision and 34.3% for recall with the 100 instance test set. The result shows 95.2% for precision and 37.6% for recall with the 200 instance test set. Although this is a pilot study, this experiment shows a promising result for Japanese particle error detection.

Key terms: Automatic Error Detection, Learner Corpora, Support Vector Machines, N-gram, Case Particle Detection

1. Introduction

The goal of the work is to automatically identify errors of case particles in Japanese learners' writing by looking at the local contextual cues around a target particle. Automatic error detection is an important task for helping to build learner corpora with error information. Learner corpora consist of language learners' spoken or written texts and are a valuable resource for reconsidering teaching methodology, materials or classroom management. There are a number of English learner corpora such as the International Corpora of Learners of English (ICLE), the Cambridge Learner Corpus (CLC), the JEFLL (Japanese EFL Learner) Corpora , and the JLE corpus (or SST corpus) that was compiled by NICT (the National Institute of Information and Communications Technology)(*Learner Corpus: Resources*, n.d.).

There are a couple of Japanese language learner corpora such as the multilingual databases of Japanese language learners' essays compiled by the National Institute of Japanese Language, which is called "Taiyaku" DB[1] and the KY corpus (Kamata & Yamauchi, 1999), compiled by a special interest group. The former consists of about 1,000 essays written by learners from 15 different countries. The latter consists of speech data from one hundred Japanese learners.

Learner corpora, different from other types of existing corpora (e.g., the British National Corpus or the Brown

Corpus), include erroneous sentences mingled with normal sentences. Because of this, it is quite a task to find those errors. To gain insights from the learner corpus and to contribute to Second Language Acquisition (SLA) research, it is necessary to detect mistakes in the learners' production, which is an extremely demanding task. Automatic error detection is difficult because there are so many error patterns to generalize. Some researchers have broken down the error detection task into certain types of errors; e.g., ill-formed spelling errors (Mays, Damerau, & Mercer, 1991; Wilcox-O'Hearn, Hirst, & Budanitsky, 2008) , mass count noun errors (Brockett, Dolan, & Gamon, 2006; Nagata et al., 2006) and preposition errors(Chodorow, Tetreault, & Han, 2007; Tetreault & Chodorow, 2008; De Felice & Pulman, 2007, 2008), because all the different types of learners' errors are too numerous to detect.

Thus, I propose an approach to learning which particle is most appropriate in a given context by representing the context as a vector populated by features referring to its syntactic characteristics. I used a machine learning algorithm known as Support Vector Machines (SVMs) with preprocessing methods to identify appropriate particle usage in a corpus of learners' writing. In the sections below, I first discuss related work on Japanese case particle error detection and then discuss the particle identification and error detection experiments and results.

2. Previous Research on Automatic Error Detection

Error detection research has been conducted for several purposes such as to check the performance of a machine translation system (Suzuki & Toutanova, 2006a, 2006b) and to check for errors in Japanese learners' writing (Imaeda, Kawai, Ishikawa, Nagata, & Masui, 2003; Nampo, Ototake, & Araki, 2007). Imaeda et al. (2003) proposed a method based on grammar rules and semantic analysis with a case frame dictionary for detection and correction for Japanese Second Language (JSL) learners' writing. In the approach based on grammar rules, it is regarded as almost impossible to write entirely flawless rules of the language models.

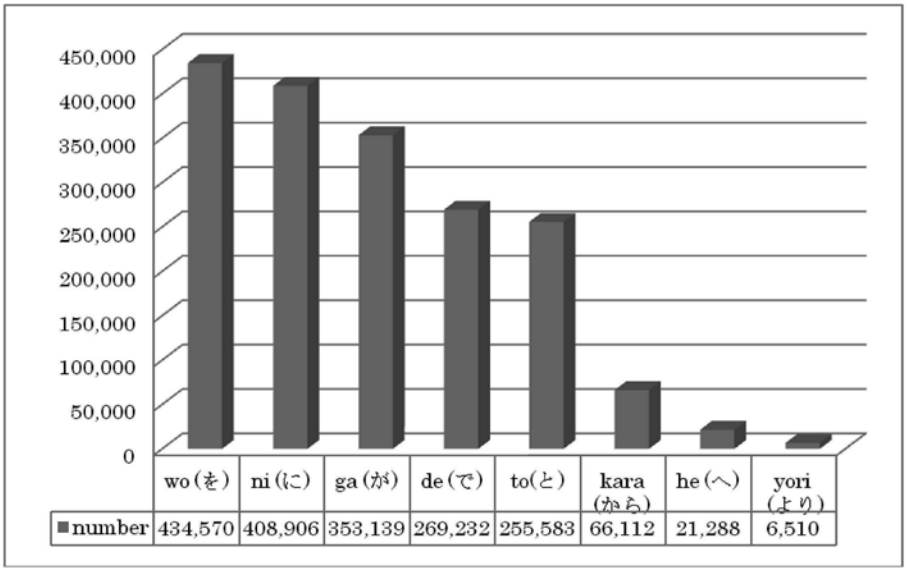
Nampo et al. (2007) also examined detection and a correction method for all of the Japanese particles (not limited to case particles) by using the clause information in a sentence. They separated a sentence into clauses and used surface forms, parts of speech (POS) for each word in the target clause, the dependent clause and clauses neighboring the target clause. For example, in a sentence "*watashi-wa ringo-mo mikan-mo sukidesu* " (I like both apples and oranges.) if the clause "*mikan-mo*" (and oranges) is taken as a target clause, then the particle or POS of information of the neighboring clause, "*ringo-mo*" (both apples…) are used as features. They reported a recall of 84% and a precision of 64% for detection, and a recall of 14% and a precision of 78% for correction. However, Nampo et al. (2007) conducted evaluation on only 84 selected sentences from learners' essays, which may be too small-scale to present an accurate assessment of its effectiveness.

As Chodorow and Leacock (2000) mention, it is difficult to build a model of incorrect usage. Thus, I considered proceeding without such a model: representing an appropriate word usage and comparing a novel example to that model. Firstly, I identify an appropriate usage model of Japanese particles and then differentiate an incorrect usage of Japanese particles by using such a model. In other words, the occurrences are identified as incorrect particle usage by using the appropriate case particle usage model.

3. Automatic Identification of Japanese Case Particles

3-1. Appropriate Case Particle Model

I conducted an experiment on extracting appropriate patterns of case particle usage from a Japanese corpus to highlight inappropriate usages because models of inappropriate usages are hard to come by. I started with the Japanese particles because particle errors are frequent in JSL writing and are likely to result in misunderstanding of a sentence. I used a newspaper corpus for creating a model that diagnoses correct use of case particles. I used eight particles: "*ga*(が)", "*wo* (を)", "*ni* (に)", "*de* (で)", "*to* (と)", "*he* (へ)", "*yori* (より)" and "*kara* (から)". Figure 1 shows the number of all case particles appearing in Mainichi-shimbun Japanese newspapers for half a year. As the figure shows, "*wo* (を)" is the most frequent, followed by "*ni* (に)", "*ga* (が)", "*de* (で)", "*to* (と)" and so forth. I selected the five most frequently occurring case particles and trained a model to choose a proper usage of a particle from the newspaper text corpus and to decide between one case particle and all other particles such as between particle "*ga* (が)" and all others, particle "*wo* (を)" and the others, and so forth.



	wo (を)	ni (に)	ga (が)	de (で)	to (と)	kara (から)	he (へ)	yori (より)
number	434,570	408,906	353,139	269,232	255,583	66,112	21,288	6,510

Figure 1: The Number of Occurrences of All Case Particles

3-2. Experimental Setup

Language Model

I used an N-gram model for sentence features to identify a correct language model. N-gram language models are based on the idea that a word (or letter) is affected by neighboring words or letters. As Firth (1957) famously states: "you shall know a word by the company it keeps (p. 11)," the collocating words are a key to learn which particle is most appropriate in a given context. If the combination of the word (or letter) appears often, there is a strong collocation relation among those words (or letters). "N" indicates the number of a word (or letter) such as N=1, 2, 3 and these are referred to as uni-gram, bi-gram and tri-gram models, respectively (Manning & Schutze, 1999)(cf. Table1). An N-gram model can predict the "N" th item by using the (N-1) th item as a condition. For example, the bi-gram language model is based on the probability of two

words (or letters) occurring together; the occurrence of a word (or letter) depends on one previous item in a certain context, which represents how strongly the two items collocate. N-gram language models are already incorporated into several studies (Kondou, 2000). I used a word-level N-gram model for error detection with the machine learning method, SVMs.

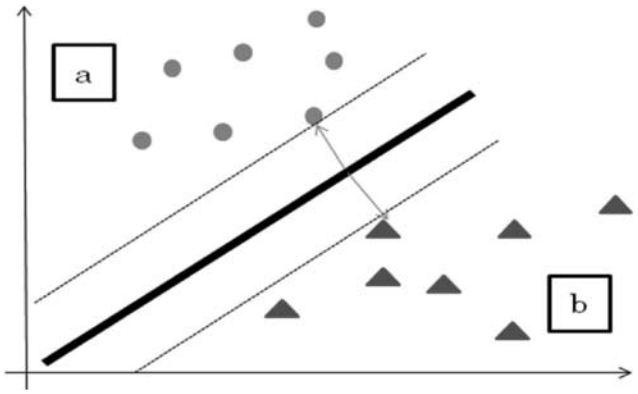


Figure 2: Image of SVMs Classification

Machine Learning Method

I used SVMs, which are methods for categorization, to train the machine learning models used in the experiments (here I used the TinySVM[2] implementation). SVMs are robust text classification methods that are widely used in the field of natural language processing, for such tasks as text classification, parts-of-speech (POS) tagging, and dependency parsing. Training examples are labeled positive or negative and tagged with features. The features are used to map each piece of data into a multi-dimensional space. If the features are similar, they are mapped closely with each other; in this way the two different classes are separated into two groups, “a” and “b” (cf. Figure2). SVMs maximize the differences between positive and negative examples; that is, the mathematical modeling is optimized to learn what the difference is between these two groups.

uni-gram (1)	“a”	“ あ ”	“sky”	“ 空 ”
bi-gram (2)	“ab”	“ あい ”	“sky is”	“ 空は ”
tri-gram (3)	“abc”	“ あいう ”	“sky is blue”	“ 空は青 ”

Table 1: Example of N-gram Collocation

training	test
10,000	1,000
50,000	5,000
100,000	10,000
200,000	20,000

Table 2: Training & Test set

Data

The data was from half-a-year's worth of articles from Mainichi-shimbun, a Japanese newspaper, from 2003, which consists of about one million characters. Sentences were first parsed with Cabocha[3], a machine learning-based Japanese syntactic dependency parser (Kudo & Matsumoto, 2002). Then, word and POS information was extracted from the words surrounding the target particles as shown in Figure 4. The data was then separated into training data and test data with a ratio of ten to one. In this experiment, I chose 10,000 instances (one instance consists of one particle with surrounding word information) for the training data and 1,000 for the test data: 50,000 for the training data and 5,000 for the test data and so on. Figure 3 shows the flow of the case particle detection experiment. After being morphologically analyzed by CaboCha, the surface form and POS information were extracted. SVMs were trained to create a language model to diagnose whether a case particle is appropriate in a given context. Each of the classifiers was tested with test data to confirm how accurate the classifier was with the metric below.

Procedure

Because SVMs optimize the difference between two groups, it is advisable to use the features that highlight the divergence between the groups. I used the following features for SVMs: 1) surface forms of words, 2) POS information within a window of ± 3 words from a case particle[4] (Kudo & Matsumoto, 2002). In Figure 4, the target case particle is "*wo* (を)" and the surface forms of the tokens such as "*nado* (such as)", "*no* (of)", "*katsudou* (activity)" are considered before "*wo* (を)", and "*sasaeru* (to support)", "*supootaa* (supporters)", "wo (particle)" are taken as the features after "*wo* (を)". In addition, POS information such as "particle（など）", "particle（の）" ,"noun（活動）" is considered before "*wo* (を)"and "verb（支える）", "noun（サポーター）", "particle（を）" are considered after "*wo* (を)". The dependency shows that the verb "*sasaeru* (to support)" determines that "*wo* (を)" is required in this sentence.

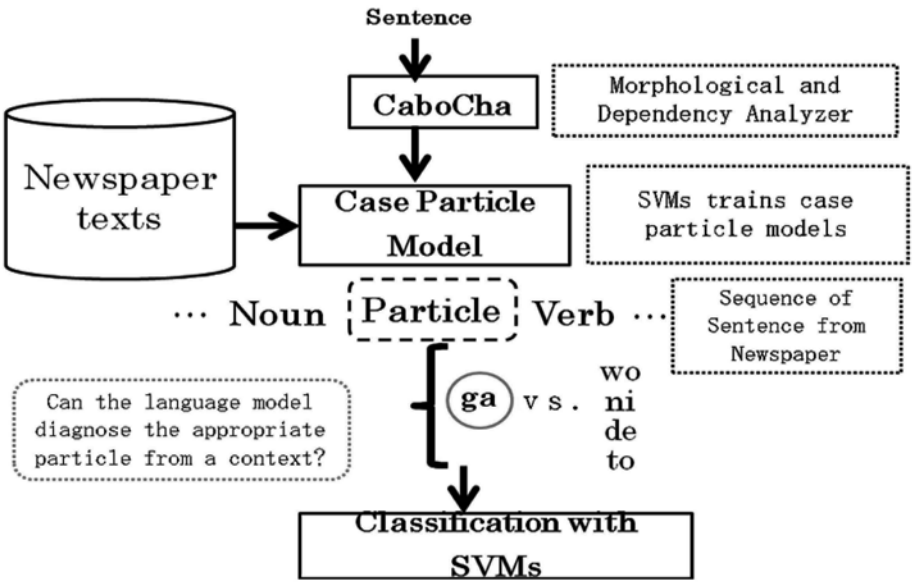


Figure 3: Flow of Case Particle Identification Experiment

Wo							
	-3	-2	-1	0	+1	+2	+3
	など	の	活動	[を]	支える	サポーター	を
Reading	nado	no	katsudou	[wo]	sasaeru	sapootaa	wo
Meaning	such-as	of	activity	[object marker]	to support	supporter	[object marker]
Parts Of Speech	Particle	Particle	Noun	[Particle]	Verb	Noun	Particle
Dependency	sasaeru (to support):Verb						
English meaning	…supporters who support activities such as ….						

Figure 4: Overview of Input Features

Metrics

Recall is the ratio of correctly retrieved instances to all target instances. Recall is calculated with the following formula.

$$Recall = \frac{Correctly\ Retrieved\ Instances}{All\ Target\ Instances}$$

Precision is the ratio of correctly retrieved instances to the instances the system determines as correct answers. Precision is calculated with the following formula.

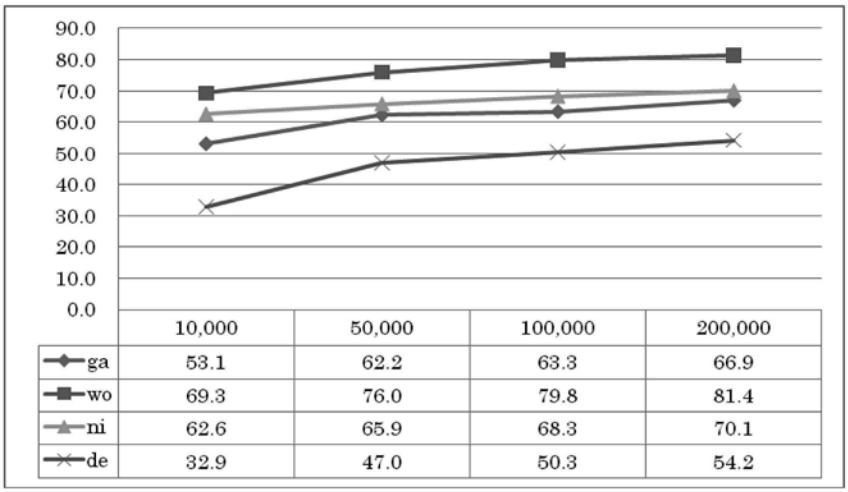
$$Precision = \frac{Correctly\ Retrieved\ Instances}{System's\ Detected\ Instances}$$

I used the F score here to observe the coordination of recall and precision scores. F score presents the overall performance of how well the system can detect the appropriate usage of particles.

$$Fscore = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Results

The result differs according to the kind of particle. The graph in Figure 5 shows the result of the experiment. The object marker "*wo* (を)" had the best score of 81.4%, 70.1% for "*ni* (に)", 66.9% for "*ga* (が)" and 54.2% for "*de* (で)" and "*to* (と)". The "*wo* (を)" is more easily detectable than the other particles, including "*to* (と)" or "*de* (で)", which have lower scores. The reason for low scores for "*de* (で)" and "*to* (と)" may reside in the fact that the models for those particles were not trained as much because they were less frequently used in the text than "*ni* (に)" or "*wo* (を)". I consider that this result depends largely on the particle frequency distribution in the corpus I used. Thus, I will equalize the number of each particle and try to minimize the disadvantages of the less-frequent particles for the next experiment.



	10,000	50,000	100,000	200,000
ga	53.1	62.2	63.3	66.9
wo	69.3	76.0	79.8	81.4
ni	62.6	65.9	68.3	70.1
de	32.9	47.0	50.3	54.2

Figure 5: Result of Case Particle Experiment

4. Automatic Error Detection with a learner corpus

Second, I performed an experiment to see how the learned model is used to detect wrong usages of particles in a learner corpus. The "*wo* (を)" model gave the best score among other particles, so I used the "*wo* (を)" particle model to detect wrong usages of "*wo* (を)" in learners' writing. The test data consist of sentences with correct and incorrect usages of "*wo* (を)". In this experiment, I used 100 and 200 instances from the learner corpus. I used the model to pick out erroneous instances from the test sets (in which there are correct and incorrect instances). In the 100-instance test set, there are 27 wrong usages and 73 correct usages of "*wo* (を)". In the 200-instance test set, there are 43 wrong usages and 157 correct usages of "*wo* (を)". The result shows 92.6% for precision and 34.3% for recall with the former test set, and 95.2% for precision and 37.6% recall for the latter.

	100	200
Precision	92.6%	95.2%
Recall	34.3%	37.6%
F score	50%	53.9%

Table 3: Result of error detection experiment of "*wo* (を)"

Here, recall is the ratio of correctly retrieved wrong usages of "*wo* (を)" to all incorrect instances, while precision is the ratio of correctly retrieved wrong usages to all retrieved instances. I found that one of the reasons for low recall results derives from the variations of Japanese writing, that is, the learners tend to use Japanese phonetic characters, hiragana, whereas kanji, or Chinese characters, are used in ordinary Japanese. The training sentences I used are extracted from Japanese newspaper articles where kanji characters are always used when appropriate. Thus, when the learners use hiragana instead of kanji, the model classifies them as wrong usages since it has never seen such usages. On the other hand, high precision shows that the model has high performance in detecting incorrect usages of particles. Below are the examples of sentences [5] that were retrieved as incorrect usages of "*wo* (を)".

1. ... けんこう｜の｜ため｜[を]｜しんじる｜。｜＜ EOS[6]＞
...believe in it for the health...
2. ... が｜この｜きせい｜[を]｜さんどうする｜。｜＜ EOS ＞
...(somebody) agrees with this regulation...
3. ... で｜も｜たばこ｜[を]｜さわる｜こと｜が｜...
...thing (that one) gets in tobacco...

5. Conclusion

In this article, I proposed an approach for detecting appropriate usage models of Japanese case particles in order to create an automatic error detection system for JSL learners' writing. The experiment resulted in different performance scores according to the kind of case particle, and the case particle "*wo* (を)" had the most significant result among all case particles. This finding may depend heavily on the number of times a particle is used in a text, "*wo* (を)" being the most frequently occurring particle in the corpus I used for this experiment, which is a factor I will take into account in future work. In future studies, I will also examine how the choice of different features affects the results and how much the appropriate model approach can help automatic case particle error detection.

Notes

1. http://www2.kokken.go.jp/eag/wiki.cgi?page=taiyakuDBn%2Ftop
2. http://chasen.org/_taku/software/TinySVMS/
3. http://chasen.org/_taku/software/cabocha/
4. A preliminary experiment with a window size of three words provided the best result among the other sizes of 4 and case particle is theoretically decided by the neighboring words and the verb information the particle depends on.
5. Kanji scripts were changed into hiragana
6. EOS means "the end of sentence".

References

Brockett, C., Dolan, W., & Gamon, M. (2006). Correting ESL Errors Using Phrasal SMT Techniques. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, 249–256.

Chodorow, M., & Leacock, C. (2000). An Unsupervised Method for Detecting Grammatical Errors. *Proceedings of the 6th Applied Natural Language Processing Conference*, 140–147.

Chodorow, M., Tetreault, J., & Han, N.-R. (2007). Detection of grammatical errors involving prepositions. *Proceedings of the 4th ACL–SIGSEM Workshop on Prepositions*, 25–30.

De Felice, R., & Pulman, S. (2007). Automatically acquiring models of prepositional use. *Proceedings of the 4th ACL–SIGSEM Workshop on Prepositions*, 45–50.

De Felice, R., & Pulman, S. (2008). A classifier-based approach to preposition and determiner error correction in L2 English. *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, 169–176.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-1955. In F.R.Palmer (Ed.), *In Studies in Linguistic Analysis* (pp. 1–32).
U.K.: Oxford:Philological Society. (Reprinted Selected Papers of J.R.Firth 1952-1959, London:Longman, 1968)

Imaeda, K., Kawai, A., Ishikawa, Y., Nagata, R., & Masui, F. (2003). Error Detection and Correction of Case Particles in Japanese Learner's Composition. *Proceedings of the Information Processing Society of Japan SIG*, 39–46.

Kamata, O., & Yamauchi, H. (1999). *KY corpus versition 1.1* (Report). Dainigengo toshiteno Nihongo no shuutoku nikansuru sougou kenkyu group (Vocaburary Acquisition Study Group).

Kondou, M. (2000). N-gram toukeishori wo mochiita mojiretsubunseki niyoru nihonkotenbungaku no kenkyu. *Jinbun kenkyu:University of Chiba*, *29*.

Kudo, T., & Matsumoto, Y. (2002). Japanese dependency analyisis using cascaded chunking. *Proceedings of the 6th Conference on Computational Natural Language Learning (CoNLL)*, 63–69.

Manning, C., & Schutze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, U.K.: The MIT Press.

Mays, E., Damerau, F., & Mercer, R. (1991). Context based spelling correction. *Information Processing and Management*, *23*(5),517–522.

Nagata, R., Wakana, T., Kawai, A., Morihiro, K., Masui, F., & Isu, N. (2006). Kasan Hukasan no Hantei ni Motozuita Eibun no Ayamarikenshutsu [Recognizing errors in english writing based on the mass count distinction]. *Proceedings of the Institute of Electronics, Information and Communication Engineers*, *J89-D*(8), 1777–1790.

Nampo, R., Ototake, H., & Araki, K. (2007). Bunsetunai no tokutyo wo motiita nihongo josi ayamari no jidoukennshutu to kousei [Automatic Error Detection and Correction of Japanese Particles Using Features within Bunsetsu]. *Proceedings of the Information Processing Society of Japan SIG*, 107–112.

Suzuki, H., & Toutanova, K. (2006a). Kikaigakushu niyoru Nihongo Kakujoshi no Yosoku [Prediction of Japanese Case Markers Using Machine Learning Methods]. *Proceedings of the 12th Annual Meeting of the Society of Natural Language Processing*,1119–1122.

Suzuki, H., & Toutanova, K. (2006b). Learning to Predict Case Makers in Japanese. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics(ACL)*,1049–1056.

Tetreault, J., & Chodorow, M. (2008). The Ups and Downs of Preposition Error Detection in ESL Writing. *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*.

Tono, Y. *Learner Corpus: Resources.* (n.d.). Retrieved June 15, 2009, from http://leo.meikai.ac.jp/_tono/

Wilcox-O'Hearn, A., Hirst, G., & Budanitsky, A. (2008). Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model. *Proceedings of 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2008)(Lecture Notes in Computer Science (CICLing-2008)(Lecture Notes in Computer Science Vol.4919)*, 605–616.